BART: A Modular Toolkit for Coreference Resolution

Yannick Versley University of Tübingen versley@sfs.uni-tuebingen.de
Simone Paolo Ponzetto EML Research gGmbH ponzetto@eml-research.de
Massimo Poesio University of Essex poesio@essex.ac.uk
Vladimir Eidelman Columbia University vae2101@columbia.edu

Alan Jern UCLA ajern@ucla.edu
Jason Smith Johns Hopkins University jsmith@jhu.edu
Xiaofeng Yang Inst. for Infocomm Research xiaofengy@i2r.a-star.edu.sg
Alessandro Moschitti University of Trento moschitti@dit.unitn.it

Abstract

Developing a full coreference system able to run all the way from raw text to semantic interpretation is a considerable engineering effort, yet there is very limited availability of off-the shelf tools for researchers whose interests are not in coreference, or for researchers who want to concentrate on a specific aspect of the problem. We present BART, a highly modular toolkit for developing coreference applications. In the Johns Hopkins workshop on using lexical and encyclopedic knowledge for entity disambiguation, the toolkit was used to extend a reimplementation of the Soon et al. (2001) proposal with a variety of additional syntactic and knowledge-based features, and experiment with alternative resolution processes, preprocessing tools, and classifiers.

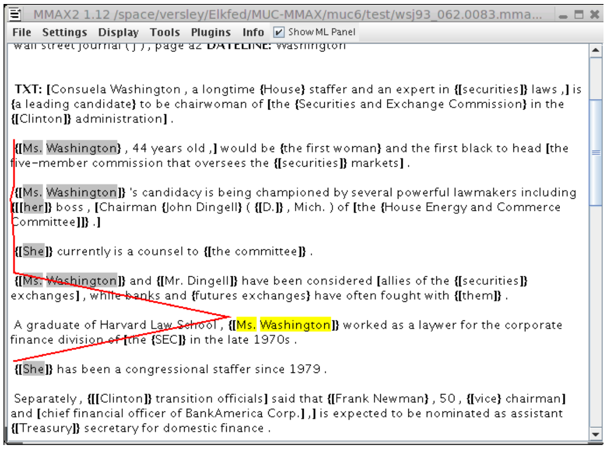
1 Introduction

Coreference resolution refers to the task of identifying noun phrases that refer to the same extralinguistic entity in a text. Using coreference information has been shown to be beneficial in a number of other tasks, including information extraction (McCarthy and Lehnert, 1995), question answering (Morton, 2000) and summarization (Steinberger et al., 2007). Developing a full coreference system, however, is a considerable engineering effort, which is why a large body of research concerned with feature engineering or learning methods (e.g. Culotta et al. 2007; Denis and Baldridge 2007) uses a simpler but non-realistic setting, using pre-identified mentions, and the use of coreference information in summarization or question answering techniques is not as widespread as it could be. We believe that the availability of a modular toolkit for coreference will significantly lower the entrance barrier for researchers interested in coreference resolution, as well as provide a component that can be easily integrated into other NLP applications.

A number of systems that perform coreference resolution are publicly available, such as GUITAR (Steinberger et al., 2007), which handles the full coreference task, and JAVARAP (Qiu et al., 2004), which only resolves pronouns. However, literature on coreference resolution, if providing a baseline, usually uses the algorithm and feature set of Soon et al. (2001) for this purpose.

Using the built-in maximum entropy learner with feature combination, BART reaches 65.8% F-measure on MUC6 and 62.9% F-measure on MUC7 using Soon et al.'s features, outperforming JAVARAP on pronoun resolution, as well as the Soon et al. reimplementation of Uryupina (2006). Using a specialized tagger for ACE mentions and an extended feature set including syntactic features (e.g. using tree kernels to represent the syntactic relation between anaphor and antecedent, cf. Yang et al. 2006), as well as features based on knowledge extracted from Wikipedia (cf. Ponzetto and Smith, in preparation), BART reaches state-of-the-art results on ACE-2. Table 1 compares our results, obtained using this extended feature set, with results from Ng (2007). Pronoun resolution using the extended feature set gives 73.4% recall, coming near specialized pronoun resolution systems such as (Denis and Baldridge, 2007).

Proceedings of the ACL-08: HLT Demo Session (Companion Volume), pages 9–12,
Columbus, June 2008.


MMAX2 1.12 /space/versley/Elkfed/MUC-MMAX/muc6/test/wsj93_062.0083.mma...
File
Settings
Display
Tools
Plugins
Info
☑ Show ML Panel

wall street journal (J) , page a2 **DATELINE:** Washington

TXT: [Consuela Washington , a longtime {House} staffer and an expert in {[securities]} laws ,] is {a leading candidate} to be chairwoman of [the {Securities and Exchange Commission} in the {[Clinton]} administration] .

{[Ms. Washington} , 44 years old ,] would be {the first woman} and the first black to head [the five−member commission that oversees the {[securities]} markets] .

{[Ms. Washington]} 's candidacy is being championed by several powerful lawmakers including {[[her]} boss , [Chairman {John Dingell} ({[D.]} , Mich.) of [the {House Energy and Commerce Committee]]} .]

{[She]} currently is a counsel to {[the committee]} .

{[Ms. Washington]} and {[Mr. Dingell]} have been considered [allies of the {[securities]} exchanges] , while banks and {futures exchanges} have often fought with {[them]} .

A graduate of Harvard Law School , {[Ms. Washington]} worked as a laywer for the corporate finance division of [the {SEC]} in the late 1970s .

{[She]} has been a congressional staffer since 1979 .

Separately , {[[Clinton]} transition officials] said that {[Frank Newman} , 50 , {[vice} chairman] and [chief financial officer of BankAmerica Corp.] ,] is expected to be nominated as assistant {[Treasury]} secretary for domestic finance .

Figure 1: Results analysis in MMAX2

2 System Architecture

The BART toolkit has been developed as a tool to explore the integration of knowledge-rich features into a coreference system at the Johns Hopkins Summer Workshop 2007. It is based on code and ideas from the system of Ponzetto and Strube (2006), but also includes some ideas from GUITAR (Steinberger et al., 2007) and other coreference systems (Versley, 2006; Yang et al., 2006). [1]

The goal of bringing together state-of-the-art approaches to different aspects of coreference resolution, including specialized preprocessing and syntax-based features has led to a design that is very modular. This design provides effective separation of concerns across several several tasks/roles, including engineering *new features* that exploit different sources of knowledge, designing improved or specialized *preprocessing* methods, and improving the way that coreference resolution is mapped to a *machine learning* problem.

Preprocessing To store results of preprocessing components, BART uses the standoff format of the MMAX2 annotation tool (Müller and Strube, 2006) with MiniDiscourse, a library that efficiently implements a subset of MMAX2's functions. Using a generic format for standoff annotation allows the use of the coreference resolution as part of a larger system, but also performing qualitative error analysis using integrated MMAX2 functionality (annotation diff, visual display).

Preprocessing consists in marking up noun chunks and named entities, as well as additional information such as part-of-speech tags and merging these information into markables that are the starting point for the mentions used by the coreference resolution proper.

Starting out with a **chunking pipeline**, which uses a classical combination of tagger and chunker, with the Stanford POS tagger (Toutanova et al., 2003), the YamCha chunker (Kudoh and Matsumoto, 2000) and the Stanford Named Entity Recognizer (Finkel et al., 2005), the desire to use richer syntactic representations led to the development of a **parsing pipeline**, which uses Charniak and Johnson's reranking parser (Charniak and Johnson, 2005) to assign POS tags and uses base NPs as chunk equivalents, while also providing syntactic trees that can be used by feature extractors. BART also supports using the Berkeley parser (Petrov et al., 2006), yielding an easy-to-use Java-only solution.

To provide a better starting point for mention detection on the ACE corpora, the **Carafe pipeline** uses an ACE mention tagger provided by MITRE (Wellner and Vilain, 2006). A specialized merger then discards any base NP that was not detected to be an ACE mention.

To perform coreference resolution proper, the mention-building module uses the markables created by the pipeline to create mention objects, which provide an interface more appropriate for coreference resolution than the MiniDiscourse markables. These objects are grouped into equivalence classes by the resolution process and a coreference layer is written into the document, which can be used for detailed error analysis.

Feature Extraction BART's default resolver goes through all mentions and looks for possible antecedents in previous mentions as described by Soon et al. (2001). Each pair of anaphor and candidate is represented as a `PairInstance` object, which is enriched with classification features by feature extractors, and then handed over to a machine learning-based classifier that decides, given the features, whether anaphor and candidate are coreferent or not. Feature extractors are realized as separate classes, allowing for their independent develop-

[1] An open source version of BART is available from http://www.sfs.uni-tuebingen.de/~versley/BART/.

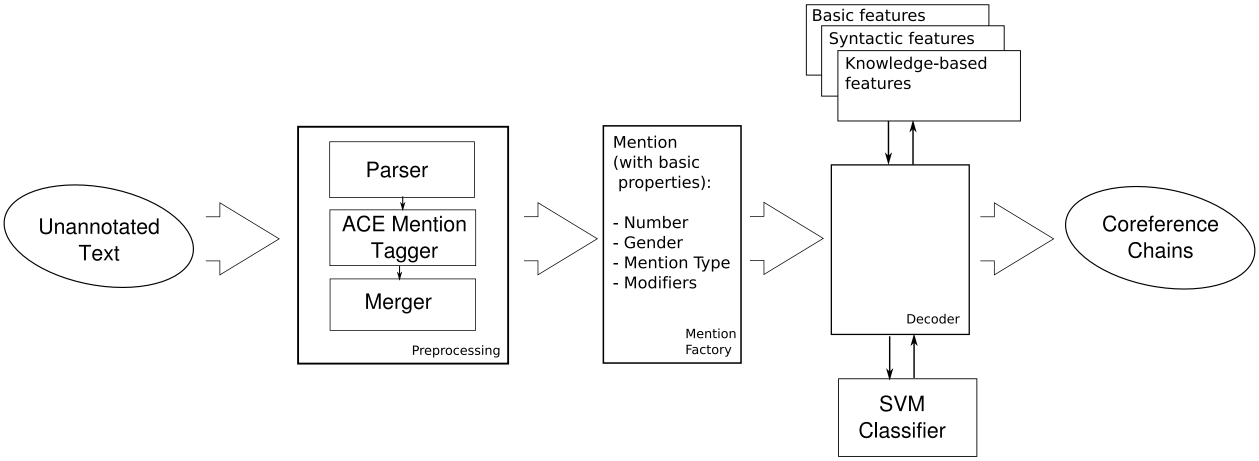


Figure 2: Example system configuration

ment. The set of feature extractors that the system uses is set in an XML description file, which allows for straightforward prototyping and experimentation with different feature sets.

Learning BART provides a generic abstraction layer that maps application-internal representations to a suitable format for several machine learning toolkits: One module exposes the functionality of the the **WEKA** machine learning toolkit (Witten and Frank, 2005), while others interface to specialized state-of-the art learners. **SVMLight** (Joachims, 1999), in the SVMLight/TK (Moschitti, 2006) variant, allows to use tree-valued features. SVM Classification uses a Java Native Interface-based wrapper replacing SVMLight/TK's `svm_classify` program to improve the classification speed. Also included is a **Maximum entropy** classifier that is based upon Robert Dodier's translation of Liu and Nocedal's (1989) L-BFGS optimization code, with a function for programmatic feature combination.[2]

Training/Testing The training and testing phases slightly differ from each other. In the training phase, the pairs that are to be used as training examples have to be selected in a process of sample selection, whereas in the testing phase, it has to be decided which pairs are to be given to the decision function and how to group mentions into equivalence relations given the classifier decisions.

This functionality is factored out into the *encoder/decoder* component, which is separate from feature extraction and machine learning itself. It is possible to completely change the basic behavior of the coreference system by providing new encoders/decoders, and still rely on the surrounding infrastructure for feature extraction and machine learning components.

3 Using BART

Although BART is primarily meant as a platform for experimentation, it can be used simply as a coreference resolver, with a performance close to state of the art. It is possible to import raw text, perform preprocessing and coreference resolution, and either work on the MMAX2-format files, or export the results to arbitrary inline XML formats using XSL stylesheets.

Adapting BART to a new coreferentially annotated corpus (which may have different rules for mention extraction – witness the differences between the annotation guidelines of MUC and ACE corpora) usually involves fine-tuning of mention creation (using pipeline and MentionFactory settings), as well as the selection and fine-tuning of classifier and features. While it is possible to make radical changes in the preprocessing by re-engineering complete pipeline components, it is usually possible to achieve the bulk of the task by simply mixing and matching existing components for preprocessing and feature extraction, which is possible by modifying only configuration settings and an XML-

[2] see `http://riso.sourceforge.net`

	BNews			NPaper			NWire		
	Recl	Prec	F	Recl	Prec	F	Recl	Prec	F
basic feature set	0.594	0.522	0.556	0.663	0.526	0.586	0.608	0.474	0.533
extended feature set	0.607	0.654	0.630	0.641	0.677	0.658	0.604	0.652	0.627
Ng 2007*	0.561	0.763	0.647	0.544	0.797	0.646	0.535	0.775	0.633

*: “expanded feature set” in Ng 2007; Ng trains on the entire ACE training corpus.

Table 1: Performance on ACE-2 corpora, basic vs. extended feature set

based description of the feature set and learner(s) used.

Several research groups focusing on coreference resolution, including two not involved in the initial creation of BART, are using it as a platform for research including the use of new information sources (which can be easily incorporated into the coreference resolution process as features), different resolution algorithms that aim at enhancing global coherence of coreference chains, and also adapting BART to different corpora. Through the availability of BART as open source, as well as its modularity and adaptability, we hope to create a larger community that allows both to push the state of the art further and to make these improvements available to users of coreference resolution.

Acknowledgements We thank the CLSP at Johns Hopkins, NSF and the Department of Defense for ensuring funding for the workshop and to EML Research, MITRE, the Center for Excellence in HLT, and FBK-IRST, that provided partial support. Yannick Versley was supported by the Deutsche Forschungsgesellschaft as part of SFB 441 “Linguistic Data Structures”; Simone Paolo Ponzetto has been supported by the Klaus Tschira Foundation (grant 09.003.2004).

References

Charniak, E. and Johnson, M. (2005). Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proc. ACL 2005*.

Culotta, A., Wick, M., and McCallum, A. (2007). First-order probabilistic models for coreference resolution. In *Proc. HLT/NAACL 2007*.

Denis, P. and Baldridge, J. (2007). A ranking approach to pronoun resolution. In *Proc. IJCAI 2007*.

Finkel, J. R., Grenager, T., and Manning, C. (2005). Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proc. ACL 2005*, pages 363–370.

Joachims, T. (1999). Making large-scale SVM learning practical. In Schölkopf, B., Burges, C., and Smola, A., editors, *Advances in Kernel Methods - Support Vector Learning*.

Kudoh, T. and Matsumoto, Y. (2000). Use of Support Vector Machines for chunk identification. In *Proc. CoNLL 2000*.

Liu, D. C. and Nocedal, J. (1989). On the limited memory method for large scale optimization. *Mathematical Programming B*, 45(3):503–528.

McCarthy, J. F. and Lehnert, W. G. (1995). Using decision trees for coreference resolution. In *Proc. IJCAI 1995*.

Morton, T. S. (2000). Coreference for NLP applications. In *Proc. ACL 2000*.

Moschitti, A. (2006). Making tree kernels practical for natural language learning. In *Proc. EACL 2006*.

Müller, C. and Strube, M. (2006). Multi-level annotation of linguistic data with MMAX2. In Braun, S., Kohn, K., and Mukherjee, J., editors, *Corpus Technology and Language Pedagogy: New Resources, New Tools, New Methods*. Peter Lang, Frankfurt a.M., Germany.

Ng, V. (2007). Shallow semantics for coreference resolution. In *Proc. IJCAI 2007*.

Petrov, S., Barett, L., Thibaux, R., and Klein, D. (2006). Learning accurate, compact, and interpretable tree annotation. In *COLING-ACL 2006*.

Ponzetto, S. P. and Strube, M. (2006). Exploiting semantic role labeling, WordNet and Wikipedia for coreference resolution. In *Proc. HLT/NAACL 2006*.

Qiu, L., Kan, M.-Y., and Chua, T.-S. (2004). A public reference implementation of the RAP anaphora resolution algorithm. In *Proc. LREC 2004*.

Soon, W. M., Ng, H. T., and Lim, D. C. Y. (2001). A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544.

Steinberger, J., Poesio, M., Kabadjov, M., and Jezek, K. (2007). Two uses of anaphora resolution in summarization. *Information Processing and Management*, 43:1663–1680. Special issue on Summarization.

Toutanova, K., Klein, D., Manning, C. D., and Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proc. NAACL 2003*, pages 252–259.

Uryupina, O. (2006). Coreference resolution with and without linguistic knowledge. In *Proc. LREC 2006*.

Versley, Y. (2006). A constraint-based approach to noun phrase coreference resolution in German newspaper text. In *Konferenz zur Verarbeitung Natürlicher Sprache (KONVENS 2006)*.

Wellner, B. and Vilain, M. (2006). Leveraging machine readable dictionaries in discriminative sequence models. In *Proc. LREC 2006*.

Witten, I. and Frank, E. (2005). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

Yang, X., Su, J., and Tan, C. L. (2006). Kernel-based pronoun resolution with structured syntactic knowledge. In *Proc. CoLing/ACL-2006*.